פרויקט בלמידה חישובית

מגישים:

אביב זבולוני – ת.ז. 211313333  
 אליה אטלן – ת.ז. 318757200

**מבוא:**

פרויקט זה עוסק בחיזוי מחלות. ננסה לבנות מודלים שחוזים האם: 1) אדם מסוים חולה במחלת לב כלשהי – [קישור לדאטא](https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction) 2) אדם מסוים חולה בסרטן הריאות – [קישור לדאטא](https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link) 3) אדם מסוים חולה בקורונה – [קישור לדאטא](https://www.kaggle.com/datasets/meirnizri/covid19-dataset)

עבור כל מחלה ננסה לאמן מודלים מ4 סוגים:

- Logistic Regression

- K Nearest Neighbors: KNN(unsupervised)

- Gaussian Mixture: GMM

- Adaboost על עצי החלטה בעומק 1 + 3 המודלים הקודמים( בחרנו את הפרמטרים האופטימליים)

עבור כל אלגוריתם נמדוד:

- הדיוק של Logistic Regression באמצעות משחק עם הפרמטר C שמגדיר את סיבוכיות המודל, ועם הפרמטר degree שמגדיר את הדרגה של הפולינום של המודל שלנו (Linear Regression)

- ה K הכי אופטימלי עבור KNN, והשוואת הדיוק של KNN אל מול האלגוריתמים האחרים באמצעות השוואת ה F1 Score.

- היכולת של GMM למצוא את מספר המחלקות הנכון (והדיוק של GMM)

- היכולת של Adaboost לשפר את הדיוק של המודלים הקיימים (נשתמש בStumps + המודלים שכבר אימנו: GMM, KNN, One Vs One)  
  
הערה: אנחנו מודעים לכך ש One Vs One הוא אלגוריתם המשמש לחיזוי K מחלקות באמצעות Logistic Regression שבנוי עבור 2 מחלקות בלבד. אבל לצורך הנוחות עם עבודה ב sklearn, נשתמש ב One Vs One לשתי מחלקות.

**הדאטא:**

**1)** **מחלת לב:**

בדאטא הזה יש 913 דוגמאות. הפעם יש רק 2 קבוצות – האם לבן אדם יש מחלת לב כלשהי, או לא. אבל יש הרבה עמודות עם מלל, לכן נהפוך אותן למספרים – למשל עמודה שמכילה את הערכים {‘Flat’, ‘Up’, ‘Down’} תתורגם ל {0,1,2}עבור כל ערך בנפרד בטבלה.  
 תחום הגילאים הוא 28-77.

**2)** **סרטן ריאות:**

בדאטא הזה יש 1000 דוגמאות. כל התכונות הם מספרים (למעט ה ID של הנבדק שאותו הסרנו). עמודת ה Y מתוארת על ידי Low, Medium, High וזה הסיכוי לחלות בסרטן ריאות. החלטנו להמיר את Low ל 1 (לא חולה) ואת Medium, High ל 1-.

תחום הגילאים הוא 14-73.

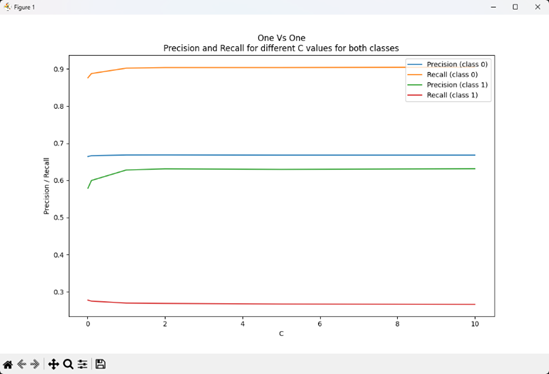
**3)** **קורונה:**

בדאטא הזה יש מליון(!) דוגמאות, מזערנו את הדאטא על ידי לקיחת 5000 שורות אקראיות.

הדאטא בא נקי וללא מלל ולכן נוכל להשתמש בו כמו שהוא (למעט השורה של התאריך שהסרנו).

תחום הגילאים הוא 0 ועד 99.

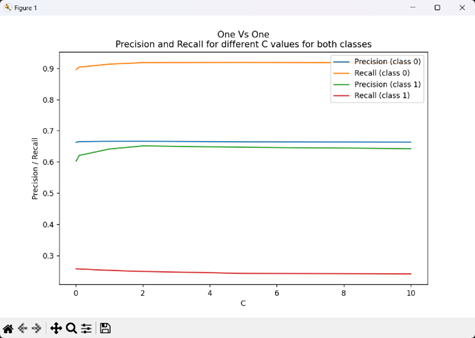
**אימון המודלים והסקת מסקנות - קורונה:**

ראשית נפריד את הדאטא לקבוצות של train, test.   
 לאחר מכן נריץ One Vs One, ונחפש את הקבוע C שמביא את התוצאות הכי טובות. התוצאות הכי טובות נמדדות כ Precision, Recall. נציג אותם בגרף הבא:

ניתן לראות כאן שה C האופטימלי הוא C = 1. כלומר סיבוכיות המודל שמתקבלת כתוצאה

מהדרישה: *C >*

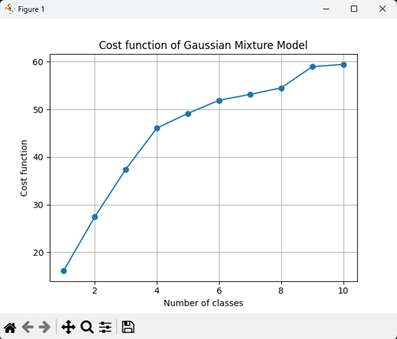
היא אופטימלית עבור הדאטא הזה.

נשים לב שה recall הוא נמוך עבור מחלקת החולים אבל גבוה עבור מחלקת הבריאים.  
 ננסה לשפר את הדרגה של התכונות ונקבל:  
  


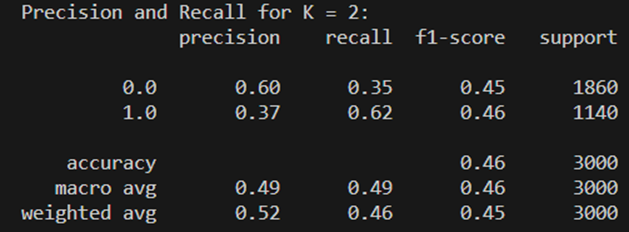
קיבלנו שיפור מזערי של ה precision של מחלקה 1.

ככל הנראה אין "מישור" מפריד בין המחלקות גם במימד גבוה יותר.

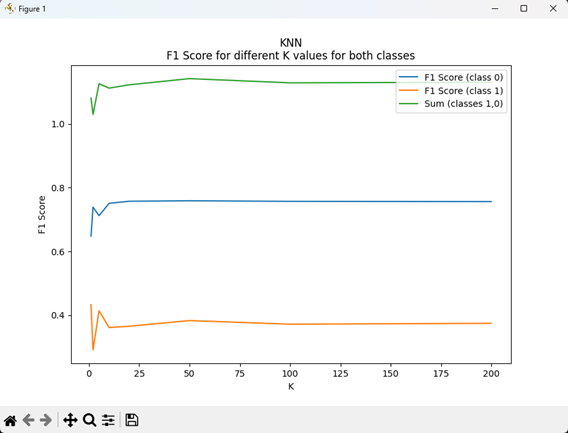
לכן במקרה הזה ניתן להסיק שהתכונות לא משפיעות הרבה על התוצאה.

נבדוק את אלגוריתם ה GMM:

נשים לב שהקפיצה הכי גדולה של פונקציית המחיר היא כאשר K = 2, ולכן מספר המחלקות האופטימלי הוא 2 (אנו יודעים שיש רק 2 מחלקות כי קיים לנו y, אך בunspervised הדבר לא נתון ואכן יש התאמה בין הK האופטימלי המצופה לK האמתי הידוע במקרה שלנו).

נבדוק את ה Recall, Precision של GMM עם K = 2:

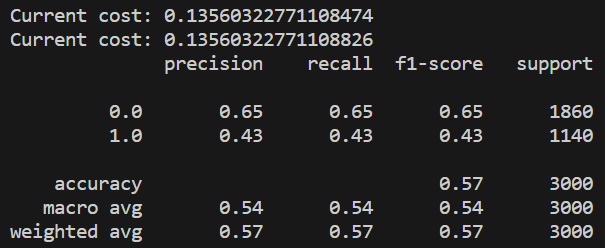
האלגוריתם הגיע לתוצאות פחות טובות גם עבור מחלקת הבריאים



נבדוק את אלגוריתם ה KNN:  
 הפעם נשווה את ה F1 Score

נשים לב שה K הכי טוב פה הוא K = 50. הקו הירוק מיצג את הסכום של שני ה scores כדי שנוכל להסיק יותר בקלות את ה K הכי טוב.

בדומה לאלגוריתם ה One Vs One, החיזוי של מחלקת הבריאים הוא מדויק ודי טוב, אבל לא טוב אל מול מחלקת החולים.

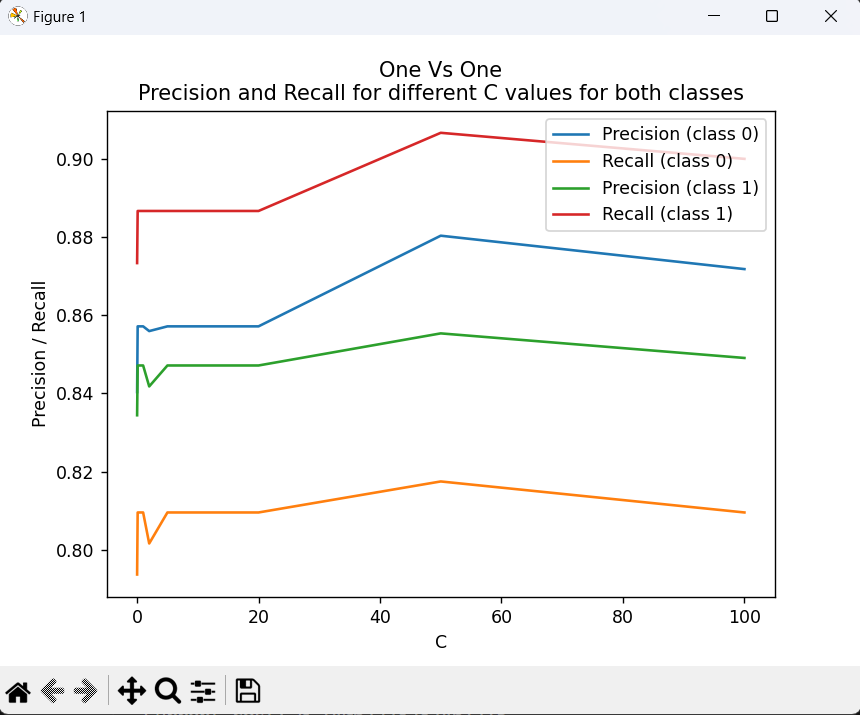


נריץ Adaboost:

האלגוריתמים שהעברנו ל Adaboost הם KNN, GMM, One Vs One

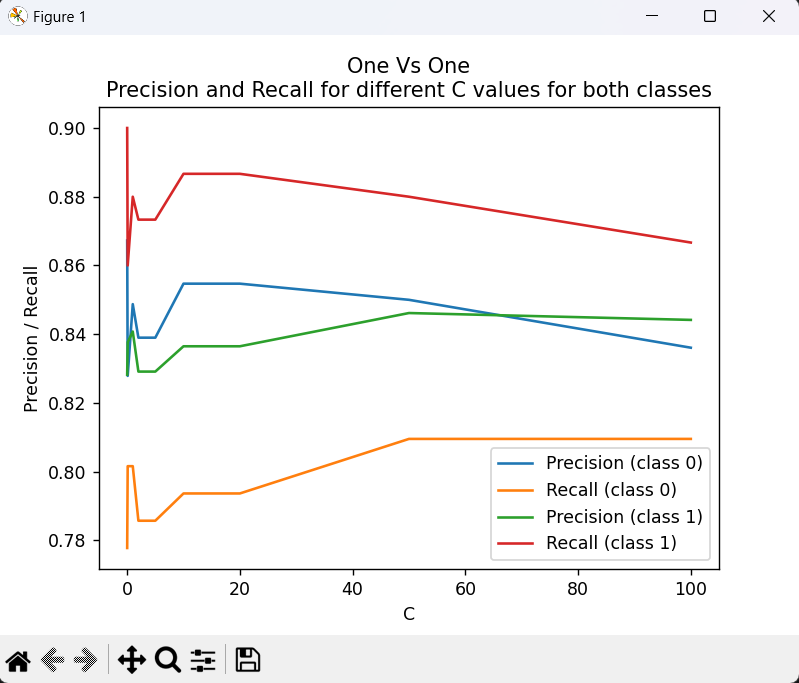
כפי שניתן לראות, השיפור העיקרי מהאלגוריתמים הקודמים הוא בעיקר לגבי מחלקת החולים  
(מחלקה 1). למרות שזה בא על חשבון הדיוק במחלקת הבריאים, נעדיף את Adaboost במקרה הזה מכיוון שלפספס גילוי של חולה זה יותר חמור מאשר לפספס גילוי של בן אדם בריא, מכיוון שבן אדם בריא יכול לעבור בדיקה נוספת, אבל אם נפספס אדם חולה, המחלה יכולה להתפשט.

לסיכום, היה קשה לעבוד עם הדאטא הזה. ככל הנראה התכונות שניתנו לא משליכות באופן משמעותי על החיזוי.

**אימון המודלים והסקת מסקנות - מחלות לב:**

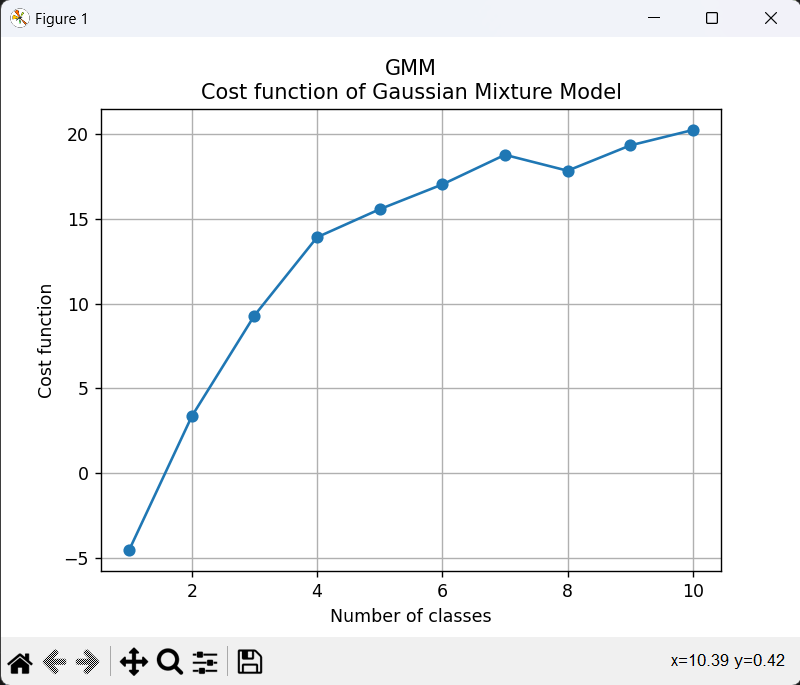
נריץ One Vs One ונחפש C אופטימלי:

ה C הכי טוב הוא 50



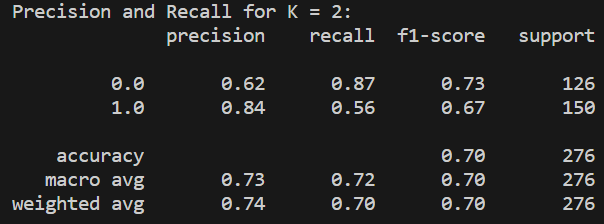
נעלה את הדרגה:

אין שיפור בכלל לאחר העלאת הדרגה, ההעלאה סיבכה את המודל יותר מידי (overfitting)



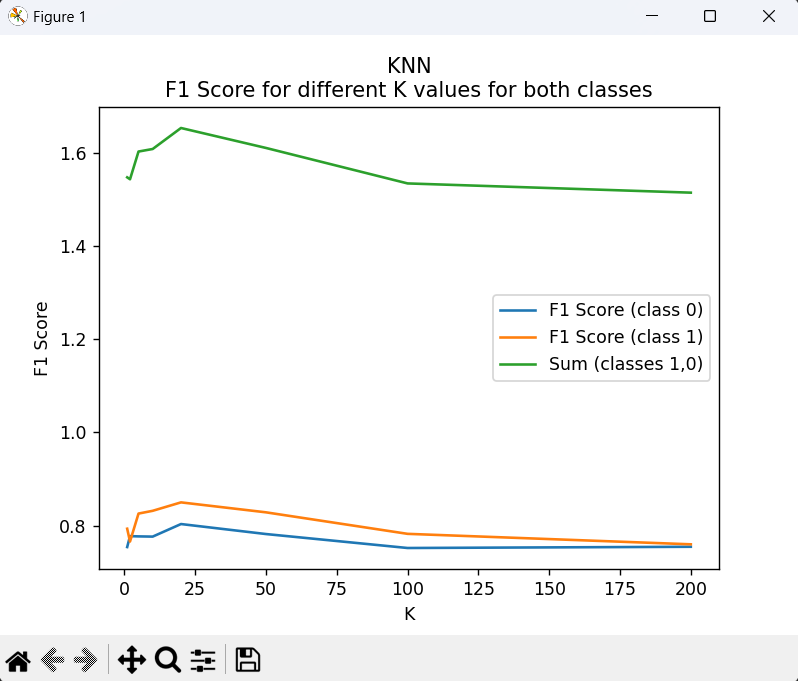
נריץ GMM:

ניתן להסיק מהגרף שמספר המחלקות הנכון הוא 2 (כפי שצפוי)



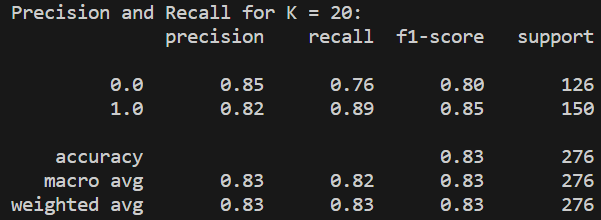
נריץ על K = 2:

נשים לב שהתוצאות לא רעות, אבל לא משתוות ל One Vs One (במידה ולא היה לנו את ה y, היינו יכולים להשתמש בו כי התוצאות לא רעות)

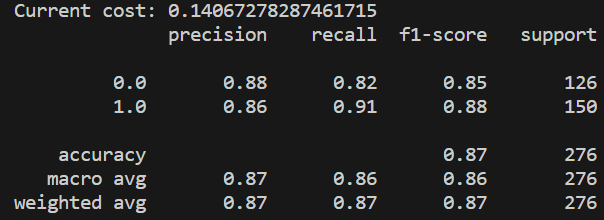


נריץ KNN ונחפש K אופטימלי:

מספר השכנים האופטימלי הוא K = 20

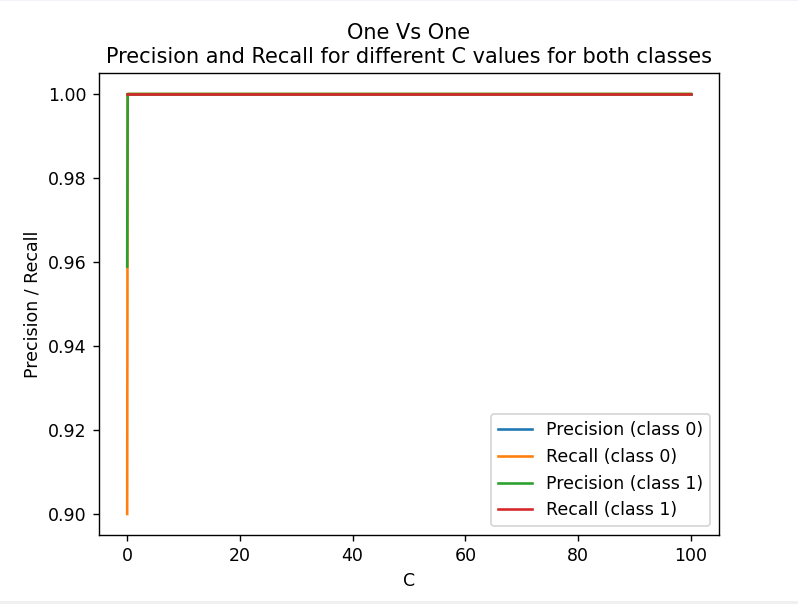


נריץ על K = 20:



נריץ Adaboost:

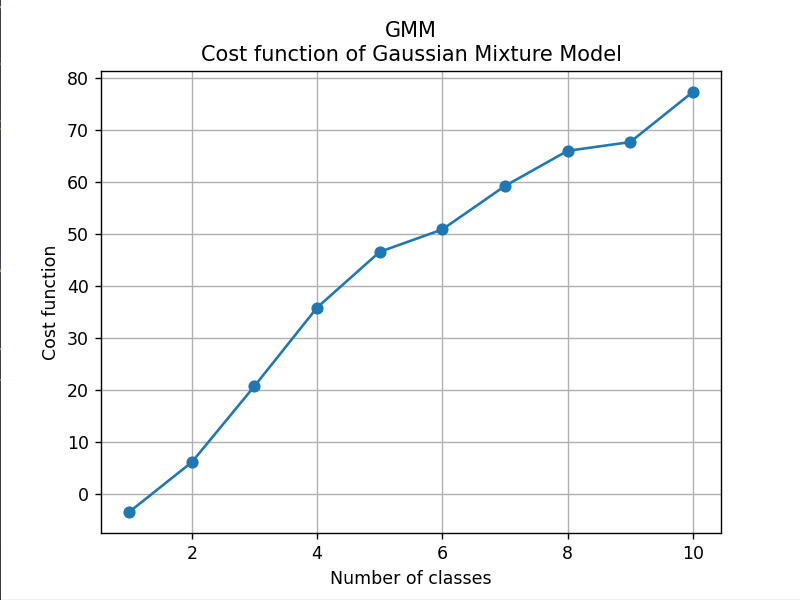
קיבלנו שיפור די טוב מ KNN, אבל תוצאות דומות ל One Vs One. (כנראה ש One Vs One מקבל משקל גדול מאוד ב Adaboost שבנינו)

**אימון המודלים והסקת מסקנות - סרטן ריאות:**

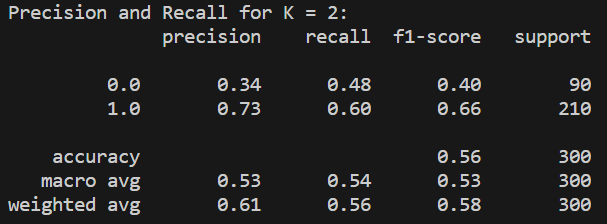
נריץ One Vs One:

נשים לב לשני דברים:

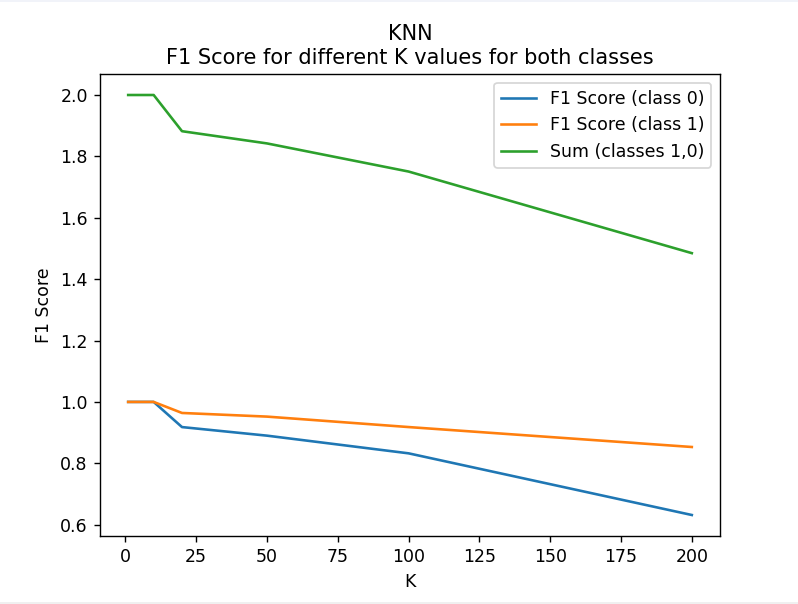
1. סיבוכיות המודל לא משנה את דיוק המודל
2. המודל מאוד מדוייק (חוזה 100% מהמחלקות, ואין שגיאה על ה test)

אין צורך להעלות בדרגה.  
נריץ GMM:

נשים לב שכל הקפיצות די דומות אחת לשנייה, והקפיצה הכי גדולה נמצאת ב K = 3.

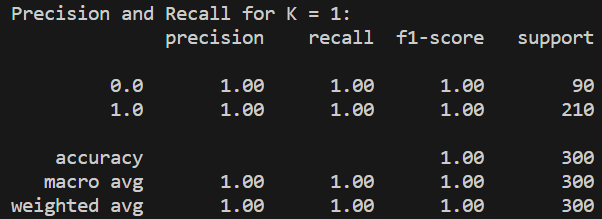
אבל אנחנו יודעים ש K = 2 ולכן נריץ GMM על K = 2:

תוצאות די גרועות, במיוחד אל מול One Vs One שעובד בצורה מושלמת



נריץ KNN:

נשים לב ש KNN פועל בצורה מושלמת כמו One Vs One על K בין 1 ל 10.



לכן נריץ KNN על K = 1 ונקבל:

אין צורך להריץ Adaboost, שכן האלגוריתמים KNN, One Vs One מגיעים לתוצאה אופטימלית על הדאטא הזה.